

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-320945

(43)Date of publication of application : 03.12.1996

(51)Int.Cl.

G06T 13/00
H04N 5/7826
H04N 5/91

(21)Application number : 07-126462 (71)Applicant : TOSHIBA CORP

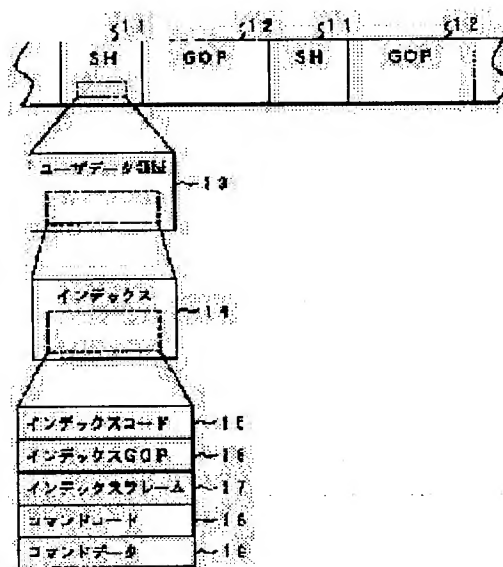
(22)Date of filing : 25.05.1995 (72)Inventor : SHIBAHARA AKIHIKO

(54) MOVING IMAGE EDITING SYSTEM AND MOVING IMAGE REPRODUCING SYSTEM

(57)Abstract:

PURPOSE: To easily edit moving image data in each frame unit and to reproduce a fine moving image from the moving image data edited in each frame unit.

CONSTITUTION: The bit stream of a moving image based upon an MPEG1 system is constituted of the repeat, of GOPs 12 consisting of plural frames and sequence headers 11 for defining the attributes of the GOPs 12. Each sequence header 11 is provided with a user data area 13. When user executes editing operation, information for defining the display order of still images in the bit stream is stored in an index 14 prepared in the user data area 13. A command code 18 expressing the sort (the insertion of another file, the partial insertion of another file, the specification of a reference file, or the specification of jump) of a command to be executed for a frame in a corresponding GOP 12 and command data 19 expression the frame or the like to be executed by the command are included in the index 14.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-320945

(43) 公開日 平成8年(1996)12月3日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 13/00			G 0 6 F 15/62	3 4 0 A
H 0 4 N 5/7826			H 0 4 N 5/782	A
5/91			5/91	N

審査請求 未請求 請求項の数 7 O L (全 15 頁)

(21) 出願番号 特願平7-126462

(22) 出願日 平成7年(1995)5月25日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 芝原 昭彦

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

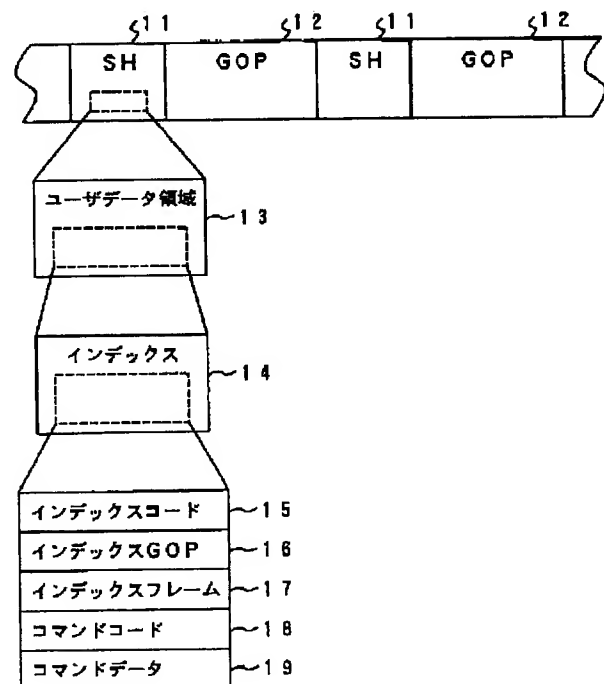
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 動画像編集システム及び動画像再生システム

(57) 【要約】

【目的】 フレーム単位での動画像データの編集を容易に行えるようにし、フレーム単位で編集された動画像データから繊細な動画像を再生できるようにする。

【構成】 M P E G 1 方式による動画像のビットストリームは、複数のフレームからなる G O P 1 2 とこの G O P 1 2 の属性を定義するシーケンスヘッダ 1 1 との繰り返しで構成される。シーケンスヘッダ 1 1 の各々は、ユーザデータ領域 1 3 が設けられている。ユーザが編集作業を行うと、ビットストリーム内の静止画像を表示する順序を定義する情報がユーザデータ領域 1 3 上にインデックス 1 4 内に記憶される。このインデックス 1 4 の中には、対応する G O P 1 2 内のフレームについて実行すべきコマンドの種類（他のファイルの挿入、他のファイルの部分挿入、参照ファイル指定、ジャンプ指定）を表すコマンドコード 1 8 及びコマンドの実行対象となるフレーム等を表すコマンドデータ 1 9 が含まれる。



【特許請求の範囲】

【請求項1】 複数の静止画像からなる画像群とこの画像群の属性を定義するヘッダとの繰り返しで構成される動画データを編集する動画編集システムにおいて、前記ヘッダに設けられるユーザデータ領域に、静止画像を表示する順序を定義する情報を含ませる手段を具備したことを特徴とする動画編集システム。

【請求項2】 前記手段がユーザデータ領域に含ませる情報の中には、処理を施すべき静止画像の識別符号及び処理の種類を表すコマンドが含まれることを特徴とする請求項1記載の動画編集システム。

【請求項3】 前記手段によりユーザデータ領域に含まれるコマンドは、挿入、削除、ジャンプ、複写のうちのいずれかであることを特徴とする請求項2記載の動画編集システム。

【請求項4】 複数の静止画像からなるGOP (group of pictures) と前記GOPの属性を定義するシーケンスヘッダとの繰り返しで構成されるMPEG1 (Motion Picture Experts Group 1) 方式のビットストリームを画面上で編集する動画編集システムにおいて、前記画面上でユーザが指定する静止画像の表示順序に基づき、前記シーケンスヘッダに設けられるユーザデータ領域に、当該シーケンスヘッダに対応するGOP内の静止画像に対して実行すべきコマンドを記憶させる手段を具備したことを特徴とする動画編集システム。

【請求項5】 前記手段によりユーザデータ領域に記憶されるコマンドは、挿入、削除、ジャンプ、複写のうちのいずれかであることを特徴とする請求項4記載の動画編集システム。

【請求項6】 複数の静止画像からなる画像群とこの画像群の属性を定義するヘッダとの繰り返しで構成される動画データから動画を再生する動画再生システムにおいて、前記ヘッダに設けられるユーザデータ領域には、静止画像を表示する順序を定義する情報が含まれており、前記ユーザデータ領域に含まれる情報に従って静止画像を表示する手段を具備したことを特徴とする動画再生システム。

【請求項7】 複数の静止画像からなるGOP (group of pictures) と前記GOPの属性を定義するシーケンスヘッダとの繰り返しで構成されるMPEG1 (Motion Picture Experts Group 1) 方式のビットストリームから動画を再生する動画再生システムにおいて、前記シーケンスヘッダに設けられるユーザデータ領域には、ユーザが画面上で指定した静止画像の表示順序に基づいて当該シーケンスヘッダに対応するGOP内の静止画像に対して実行すべきコマンドが記憶されており、前記ユーザデータ領域に記憶されたコマンドを検索して実行する手段と、

前記手段によりコマンドの実行が完了した静止画像を表

示する手段とを具備したことを特徴とする動画再生システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、動画編集システム及び動画再生システムに関し、特に、MPEG1方式による動画データを編集するための動画編集システム及びその編集された動画データから動画を再生するための動画再生システムに関する。

【0002】

【従来の技術】 近年、動画データの編集用ソフトウェアの進歩により、パーソナルコンピュータ上で手軽に動画データの編集が行われるようになってきた。ユーザは、動画を再生する際に各静止画像が所望の順序で表示されていくように編集を行う。この編集は、パーソナルコンピュータのディスプレイに表示される編集画面に向かってユーザがマウス等を操作することにより行われる。具体的には、例えばマイクロソフト社の“MS-Windows”に代表されるウインドウソフト (window software) による環境のもとで、ユーザは、インテル社の“Indeo”やマイクロソフト社の“MS-Video”といった動画編集用ソフトウェアを起動させることにより動画データの編集を行う。

【0003】 動画データは、所定の動画圧縮規格に基づいて蓄積メディア上に記録されている。例えば“MS-Windows”上で扱われる動画データは、マイクロソフト社が開発したAVI (audio video interleaving) と呼ばれるファイル形式に基づいて記録される。一方、動画圧縮規格には、ISO (国際標準化機構) 及びIEC (国際電気標準会議) により国際標準とされているMPEG1 (moving picture experts group 1) に基づくものがある。MPEG1の形式は、高い圧縮率で動画データを記録できるという特長を持つ。

【0004】 MPEG1形式による動画データは、図18に示すように、シーケンスヘッダ (sequence header) 1とGOP (group of pictures) 2の組が繰り返し現れるビットストリームを形成している。GOP2は、再生する際の画像の最小単位であるフレーム (ピクチャ、静止画像) を複数枚含んでいる。シーケンスヘッダ1は、対応するGOP2の先頭に付されており、当該GOP2におけるフレームの属性 (フォーマット等) を定義する。シーケンスヘッダがGOPごとに付されているのは、途中からの動画の再生を可能とするためである。

【0005】 圧縮された状態で記録されている動画データを編集するためには、まず、動画データを取り出してからその動画データの伸張を行う。次に、ユーザは、伸張した動画データに対して編集を施す。編集を施した後は、その動画データは再び圧縮されて蓄積メディア上に記録される。そして、編集後の動画データ

がデコード・表示され、動画像が再生されることになる。

【0006】MPEG1形式による動画像データは、編集される際にGOP単位でその伸張や圧縮がなされ、また、GOP単位でその挿入、複写、削除等の編集処理がなされる。

【0007】

【発明が解決しようとする課題】しかし、従来の動画像編集システムによれば、MPEG1形式の動画像データに代表されるように、動画像データの編集はGOP単位でなされるため、例えばGOPの途中（すなわち、フレーム間）に他のフレームを挿入したり、所定のフレームだけを削除したりするようなフレーム単位での細かな編集が困難となっていた。

【0008】さらに、従来の動画像編集システムではフレーム単位での細かな編集が困難であるため、動画像を再生する際には、ユーザが望んでいた通りの繊細な動画像が得られるとはいえなかった。

【0009】本発明は上記実情に鑑みてなされたもので、フレーム単位で容易に動画像データの編集を行うことのできる動画像編集システム及びフレーム単位で編集された動画像データから繊細な動画像を再生することのできる動画像再生システムを提供することを目的とする。

【0010】

【課題を解決するための手段】

（第1の発明）本発明に係る動画像編集システムは、複数の静止画像からなる画像群とこの画像群の属性を定義するヘッダとの繰り返しで構成される動画像データを編集する動画像編集システムにおいて、前記ヘッダに設けられるユーザデータ領域に、静止画像を表示する順序を定義する情報を含ませる手段を具備したことを特徴とする。

【0011】（第2の発明）本発明に係る動画像再生システムは、複数の静止画像からなる画像群とこの画像群の属性を定義するヘッダとの繰り返しで構成される動画像データから動画像を再生する動画像再生システムにおいて、前記ヘッダに設けられるユーザデータ領域には、静止画像を表示する順序を定義する情報が含まれており、前記ユーザデータ領域に含まれる情報に従って静止画像を表示する手段を具備したことを特徴とする。

【0012】

【作用】

（第1の発明）動画像のビットストリームは、複数の静止画像からなる画像群とこの画像群の属性を定義するシーケンスヘッダとの繰り返しで構成されている。また、前記シーケンスヘッダには、ユーザデータ領域が設けられている。本発明に係る動画像編集システムによれば、この動画像のビットストリームを編集する際に、前記シーケンスヘッダに設けられるユーザデータ領域に、静止

画像を表示する順序を定義する情報が含まれる。上記構成とすることにより、動画像データの静止画像単位での編集を簡単に行うことができる。

【0013】（第2の発明）動画像のビットストリームは、複数の静止画像からなる画像群とこの画像群の属性を定義するシーケンスヘッダとの繰り返しで構成されている。また、前記シーケンスヘッダには、ユーザデータ領域が設けられている。本発明に係る動画像再生システムによれば、前記ユーザデータ領域には静止画像を表示する順序を定義する情報が編集作業によって既に含まれており、表示手段は前記ユーザデータ領域に含まれる情報に従って静止画像を表示する。上記構成とすることにより、静止画像単位で編集された動画像データから繊細な動画像を再生することができる。

【0014】

【実施例】以下、図面を参照して本発明の一実施例を説明する。図1は、本発明に係る動画像編集システムで使用する動画像データの構成を示す図である。本実施例で扱う動画像データは、ビットストリームを形成している。ビットストリームは、図1に示すように、シーケンスヘッダ11とGOP12の組の繰り返しで構成されている。GOP12は、複数のフレーム（ピクチャ、静止画像）を1つのグループとしてまとめたものである。各シーケンスヘッダ11は、対応するGOP12に含まれるフレームの属性（フォーマット等）を定義するデータを保持している。

【0015】なお、上記ビットストリームは、後で行うビットストリームの編集がしやすくなるように、種類に応じて複数のファイルに分けて格納されている。ユーザは、任意のファイルを選択してビットストリームに含まれるフレームの列を編集画面上に読み込み、各フレームが所望の順序で表示されるように編集する。

【0016】ビットストリーム内のシーケンスヘッダ11には、図2に示すデータが格納されている。すなわち、シーケンスヘッダ11には、当該シーケンスヘッダの始まりを表すシーケンスヘッダコード（sequence header code）21のほか、上記フレームの水平サイズ（horizontal size）22、垂直サイズ（vertical size）23、画素アスペクト比（pel aspect ratio）24、等のデータが格納されている。また、これらデータの後方には、ユーザが任意の容量で自由に設定できるユーザデータ（user data）25を格納するための領域（ユーザデータ領域）が設けられている。本実施例では、後で詳述するように、上記ユーザデータ領域を利用することにより、フレーム単位の編集が可能となる。

【0017】一方、GOP12には、図3に示すデータが格納されている。すなわち、GOP12には、当該GOPの始まりを表すグループスタートコード（group start code）31のほか、画像の表示時間を示すタイムコード（time code）32、前方のGOPを参照すべきか

否かを表すフラグ (closed gop) 33, GOPが正しく再生できるような状態にあるが否かを表すフラグ (broken link) 34, 等のデータが格納されている。また、これらデータの後方には、ユーザが任意の容量で自由に設定できるユーザデータ (user data) 35を格納するためのユーザデータ領域13が設けられている。なお、上記シーケンスヘッダ11及びGOP12のデータ構造は、国際標準規格としてISO及びIECにより規定されているものである。

【0018】ここまで説明したユーザデータ領域13を含むシーケンスヘッダ11及びGOP12 (図1) は、PMEG1の規定に基づくビットストリームを形成している。一方、上記ユーザデータ領域13に含まれるインデックス14は、以下に説明するように、本発明に係るものである。

【0019】ユーザデータ領域13には、インデックス14が格納される。このインデックス14は、デコーダ (後述) がフレームを表示する際に、各フレームの表示順序を決めるために使用されるものである。

【0020】インデックス14には、フレームの表示順序を定義する情報が格納されている。すなわち、インデックス14には、インデックスコード (IndexCode) 15, インデックスGOP (IndexGop) 16, インデックスフレーム (IndexFrame) 17, コマンドコード (CommandCode) 18, コマンドデータ (CommandData) 19が格納される。

【0021】これら各種のデータがインデックス14に格納されている様子を図4に詳細に示す。インデックスコード15は、所属するインデックスを識別するためのコードである。インデックスGOP16は、所属するシーケンスヘッダに対応するGOPの番号を表す。インデックスフレーム17は、所属するシーケンスヘッダに対応するGOPに含まれるフレームの数を表す。すなわち、インデックスフレーム17は、対応するGOPの先頭から数えたフレームの数を表している。また、各フレームには、順番に番号が付されている。このとき、GOP内の先頭フレームの番号は0番となる。

【0022】コマンドコード18は、後で詳述するように、対応するGOPに含まれる処理対象のフレームに対してデコーダが実行すべきコマンドの種類を表す。コマンドデータ19は、後で詳述するように、処理対象となるフレーム等を指定するためのデータであり、上記コマンドコードの種類に応じたものがそれぞれ用意されている。なお、上記インデックス14に格納されている情報は、ユーザが編集画面 (後述) で編集した内容に応じて設定されるようになっている。

【0023】図5は、上記コマンドコード18の種類を示している。コマンドコード18には、“0x0001”, “0x0002”, “0x0003”, “0x0004”の4種類が用意されており、これらのうちのい

ずれか1つがインデックス14上に格納されるようになっている。

【0024】“0x0001”は、他のファイル (当該コマンドコードが所属するファイル以外のファイル) を、対応するGOPに含まれるフレーム間に挿入させる命令であることを表している。“0x0002”は、他のファイルに含まれるフレームを、対応するGOPに含まれるフレーム間に挿入させる命令であることを表している。“0x0003”は、他のファイルを参照する命令であることを表している。“0x0004”は、処理対象のフレームを、所属するファイル内のフレーム間又は他のファイル内のフレーム間にジャンプさせる命令であることを表している。

【0025】図6 (a) ~ (d) は、上記4種類のコマンドコードにそれぞれ属するコマンドデータのデータ構造を示している。図6 (a) に示すように、コマンドコード“0x0001” (他のファイルの挿入) に属するコマンドデータには、変数として、挿入先のGOP番号 (InsGop), 挿入先のフレーム番号 (InsFrame), 挿入ファイル名 (InsFile) が格納される。

【0026】また、図6 (b) に示すように、コマンドコード“0x0002” (他のファイルの部分挿入) に属するコマンドデータには、上記コマンドコード“0x0001”の場合と同様の挿入先のGOP番号 (InsGop), 挿入先のフレーム番号 (InsFrame), 挿入ファイル名 (InsFile) が格納され、これに加えて、挿入GOP番号 (NumGop), 挿入フレーム番号 (NumFrame), 挿入フレーム数 (Frames) が格納される。

【0027】また、図6 (c) に示すように、コマンドコード“0x0003” (参照ファイル指定) に属するコマンドデータには、GOP番号 (NumGop), フレーム番号 (NumFrame), 参照ファイルの種類 (Kind), 挿入ファイル名 (InsFile) が格納される。

【0028】最後に、図6 (d) に示すように、コマンドコード“0x0004” (ジャンプ指定) に属するコマンドデータには、GOP番号 (NumGop), フレーム番号 (NumFrame), ジャンプ先のGOP番号 (JGop), ジャンプ先のフレーム番号 (JFrame), ジャンプ先のファイル名 (JmpFile) が格納される。

【0029】ユーザの編集操作により、各シーケンスヘッダにおけるコマンドコード18及びコマンドデータ19が定まった後は、ビットストリームは圧縮されて蓄積メディア等に格納されることになる。

【0030】上記説明したインデックスコード15, インデックスGOP16, インデックスフレーム17, コマンドコード18, コマンドデータ19で構成される1組の情報は、1種類のコマンドを表している。したがって、後述のデコーダがこのコマンドを実行することにより、フレームの表示順序が決定される。なお、インデックス14に含まれる上記情報は1組だけとは限らない。

あるGOP内のフレームについて複数のコマンドを実行させる場合には、複数組の情報がインデックス14に格納される。

【0031】次に、図7を参照して、ユーザがビットストリームについて編集を行うための編集画面の構成を説明する。ユーザは、図7に示す編集画面に向かい、マウスやキーボード（図示せず）を用いて当該編集画面に現れるビットストリーム内のフレームの列を参照して編集を行う。

【0032】編集画面は、作業エリア4aと作業エリア4bにより構成される。作業エリア4aと作業エリア4bには、それぞれ異なるファイル内ビットストリームを編集するための領域である。

【0033】例えば、所定のファイル内ビットストリームに含まれるフレームの一部を削除させたい場合には、作業エリア4aだけが使用される。また、所定のファイル内ビットストリームに含まれるフレームを他のファイル内のビットストリーム中に挿入させたい場合には、作業エリア4aと作業エリア4bの両方が使用される。

【0034】作業エリア4aには、処理選択枝41a、フレーム表示部42a、カーソル43a、スクロールボタン44a、45a、表示エリア46a、スライダ47aが備えられる。処理選択枝41aには、“ファイル”、“ファイル挿入”、“フレーム挿入”、“カット”、“ジャンプ”、“参照ファイル”といった各種の作業項目が設けられている。“ファイル”は、編集対象とすべき（後述のフレーム表示部42aに表示すべき）ファイルを選択するために使用される。例えば、ユーザが“ファイル”をマウス等を用いてクリックすると、所定の位置にウィンドウが現れ、各種のファイルのファイル名が表示される。こうして表示されたファイル名のうち編集対象とすべきファイルのファイル名をユーザが再びクリックすると、相当するファイルが選択されることになる。

【0035】“ファイル挿入”は、作業エリア4aにおける編集対象のファイル内のフレーム間に、作業エリア4bにおけるファイルを挿入させるために使用される。“フレーム挿入”は、作業エリア4aにおける編集対象のファイル内のフレーム間に、作業エリア4bにおけるファイル内のフレームを挿入させるために使用される。“カット”は、編集対象のファイル内のフレームを削除するために使用される。“ジャンプ”は、編集対象のファイル内のフレームを同じファイル内又は他のファイル内のフレーム間にジャンプさせるために使用される。

【0036】フレーム表示部42aは、処理選択枝41a上の“ファイル”で選択されたファイルのビットストリームに含まれるフレームの列を表示する。この場合、各フレームは、それぞれ識別番号が付された状態で表示される。

【0037】カーソル43aは、上記フレーム表示部4

2aに表示されるフレームの列の中から処理対象となる位置（フレーム間）を指定するために使用される。カーソル43aは、図9に示すように、作業項目の種類に応じて様々な形で表示される。

【0038】上記処理選択枝41aにおいて“ファイル挿入”又は“フレーム挿入”が選択された場合、カーソル43aは“↑↑”で表示される。カーソル“↑↑”は、挿入先の位置を指定するために使用される。この場合、ユーザは、作業エリア4bにおけるファイル等を挿入すべき位置（フレーム間）に、カーソル“↑↑”を移動させる。なお、“ファイル挿入”が選択されたとき、作業エリア4b内のカーソル43bは無表示となり、“フレーム挿入”が選択されたとき、カーソル43bは“S”と“E”で表示される。カーソル“S”と“E”は、挿入元のフレーム列の範囲を指定するために使用される。

【0039】上記処理選択枝41aにおいて“カット”が選択された場合、カーソル43aは“S”と“E”で表示される。カーソル“S”と“E”は、削除すべきフレーム列の範囲を指定するために使用される。この場合、ユーザは、削除する範囲の始点にカーソル“S”を移動させ、終点にカーソル“E”を移動させる。なお、このときのカーソル43bは無表示となっている。

【0040】また、“ジャンプ”が選択された場合、カーソル43aは“J”と“D”で表示される。カーソル“J”は、ジャンプを開始すべき位置（フレーム間）を指定するために使用され、カーソル“D”は、ジャンプ先とすべき位置を指定するために使用される。この場合、ユーザは、ジャンプを開始すべき位置にカーソル“J”を移動させ、ジャンプ先とすべき位置（同一ファイル内）にカーソル“D”を移動させる。なお、このときのカーソル43bは無表示となっている。

【0041】一方、同一ファイル内へのジャンプでなく他のファイルへのジャンプを行う場合は、“ジャンプ”を選択した後、後述するように作業エリア4bに当該他のファイル呼び出す。これにより、カーソル43aは“J”が表示され、カーソル43bは“D”で表示される。

【0042】スクロールボタン44a、45aは、フレーム表示部42aに表示されるフレームの列を左右にスクロールさせるために使用される。上記カーソル43a及びスクロールボタン44a、45aを操作することにより、ユーザは、処理対象となる位置をフレーム表示部42a上で指定することができる。

【0043】表示エリア46aは、スライダ47a上を左右に移動できるようになっており、左端に移動させればファイル内の先頭フレームがフレーム表示部42a上に表示され、右端に移動させればファイル内の最終フレームが表示されるようになっている。

【0044】一方、作業エリア4bには、処理選択枝4

1 b, フレーム表示部4 2 b, カーソル4 3 b, スクロールボタン4 4 b, 4 5 b, 表示エリア4 6 b, スライダ4 7 bが備えられる。処理選択枝4 1 bには, “ファイル”, “コピー”, “キャンセル”といった各種の作業項目が設けられている。“ファイル”は, 作業エリア4 aにおけるファイル編集に使用すべき(後述のフレーム表示部4 2 bに表示すべき)他のファイルを選択するために使用される。“コピー”は, 作業エリア4 aにおける編集対象のファイル内のフレーム間に, 作業エリア4 bにおける他のファイル内のフレームを複写させるために使用される。“キャンセル”は, いま行っている編集作業を無効にする場合に使用される。

【0045】フレーム表示部4 2 bは, 処理選択枝4 1 b上の“ファイル”で選択されたファイルのビットストリームに含まれるフレームの列を表示する。この場合, 各フレームは, 識別番号が付された状態で表示される。

【0046】カーソル4 3 bは, 上記フレーム表示部4 2 bに表示されるフレームの列の中から処理対象となる位置(フレーム間)を指定するために使用される。カーソル4 3 bも, 図9に示すように, 作業項目の種類に応じて様々な形で表示される。

【0047】上記処理選択枝4 1 bにおいて“コピー”が選択された場合, 上記作業エリア4 a内のカーソル4 3 aは“↑↑”で表示される。カーソル“↑↑”は, コピー先の位置を指定するために使用される。この場合, ユーザは, 作業エリア4 aにおけるファイル内のフレームを複写すべき位置(フレーム間)に, カーソル“↑↑”を移動させる。このとき, 作業エリア4 b内のカーソル4 3 bは“S”と“E”で表示される。カーソル“S”と“E”は, 複写元のフレーム列の範囲を指定するために使用される。この場合, ユーザは, 複写する範囲の始点にカーソル“S”を移動させ, 終点にカーソル“E”を移動させる。

【0048】スクロールボタン4 4 b, 4 5 bは, フレーム表示部4 2 bに表示されるフレームの列を左右にスクロールさせるために使用される。上記カーソル4 3 b及びスクロールボタン4 4 b, 4 5 bを操作することにより, ユーザは, 処理対象となる位置をフレーム表示部4 2 b上で指定することができる。

【0049】表示エリア4 6 bは, スライダ4 7 b上を左右に移動できるようになっており, 左端に移動させればファイル内の先頭フレームがフレーム表示部4 2 b上に表示され, 右端に移動させればファイル内の最終フレームが表示されるようになっている。

【0050】次に, 上記実施例の動作を説明する。いま, ユーザは, 図7に示す編集画面でビットストリームを編集しようとしている。まず, 作業エリア4 aにおいて, ユーザが処理選択枝4 1 a内の“ファイル”を選択して編集すべきファイルを指定すると, そのファイル内のビットストリームが編集画面に読み込まれ, ビットス

トリム中のフレームの列がフレーム表示部4 2 aに表示される。

【0051】編集対象のファイル内のあるフレーム間に他のファイルを挿入するためには, ユーザは, “ファイル挿入”を選択する。これにより, カーソル4 3 aは“↑↑”となり, カーソル4 3 bは無表示となる。次に, ユーザが処理選択枝4 1 b内の“ファイル”を選択して挿入元となるファイルを指定すると, 対応するファイル内のビットストリームが編集画面に読み込まれ, 当該ファイルに含まれるフレームの列がフレーム表示部4 2 bに表示される。そしてユーザは, 作業エリア4 aにおいて, 上記挿入元のファイルを挿入すべき位置(挿入先)にカーソル“↑↑”を移動させる。この後, ユーザが“ファイル挿入”を再び選択すると, インデックス14(図4)内のコマンドコード18及びコマンドデータ19の設定が行われる。

【0052】すなわち, コマンドコード18は“0x0001”(図5)に設定され, コマンドデータ19については, 各種の変数(図6の(a))が以下のように設定される。

【0053】変数“InsGop”には, カーソル“↑↑”で指定されたフレーム間を含むGOPの番号が設定される。“InsFrame”には, カーソル“↑↑”のすぐ左に位置するフレームの番号が設定される。“InsFile”には, 挿入したファイルの名称(ファイル名)が設定される。

【0054】次に, 編集対象のファイル内のフレーム間に他のファイル内のフレームを挿入するためには, ユーザは, “フレーム挿入”を選択する。これにより, カーソル4 3 aは“↑↑”となり, カーソル4 3 bは“S”と“E”になる。ユーザが, 処理選択枝4 1 b内の“ファイル”を選択して, 挿入元フレームを含むファイルを指定すると, 当該ファイル内のビットストリームが編集画面に読み込まれ, フレームの列がフレーム表示部4 2 bに表示される。そしてユーザは, 作業エリア4 bにおいて, 挿入するフレーム列の始点にカーソル“S”を移動させ, 終点にカーソル“E”を移動させる。また, ユーザは, 作業エリア4 aにおいて, 上記挿入元のフレームを挿入すべき位置(挿入先)にカーソル“↑↑”を移動させる。この後, ユーザが“フレーム挿入”を再び選択すると, インデックス14(図4)内のコマンドコード18及びコマンドデータ19の設定が行われる。

【0055】すなわち, コマンドコード18は“0x0002”(図5)に設定され, コマンドデータ19については, 各種の変数(図6の(b))が以下のように設定される。

【0056】変数“InsGop”には, カーソル“↑↑”で指定されたフレーム間を含むGOPの番号が設定される。“InsFrame”には, カーソル“↑↑”のすぐ左側に位置するフレームの番号が設定される。“NumGop”に

は、カーソル“S”で指定されたフレーム間を含むGOPの番号が設定される。“NumFrame”には、カーソル“S”のすぐ左側に位置するフレームの番号が設定される。“Frames”には、カーソル“S”から“E”までの間に含まれるフレームの数が設定される。“InsFile”には、挿入元のファイルの名称（ファイル名）が設定される。

【0057】次に、編集対象のファイル内のフレームを削除するためには、ユーザは、“カット”を選択する。これにより、カーソル43aは“S”と“E”になり、カーソル43bは無表示になる。そしてユーザは、作業エリア4aにおいて、削除するフレーム列の始点にカーソル“S”を移動させ、終点にカーソル“E”を移動させる。この後、ユーザが“カット”を再び選択すると、インデックス14（図4）内のコマンドコード18及びコマンドデータ19の設定が行われる。なお、このときのコマンドコード18及びコマンドデータ19は、後述の“ジャンプ”の設定処理を行う場合のコマンドコード18及びコマンドデータ19が使用される。

【0058】したがって、コマンドコード18は“0x0004”（図5）に設定され、コマンドデータ19については、各種の変数（図6の（d））が以下のように設定される。

【0059】変数“NumGop”には、カーソル“S”のすぐ左側に位置するフレームを含むGOPの番号が設定される。“NumFrame”には、カーソル“S”のすぐ左側に位置するフレームの番号が設定される。“JGop”には、カーソル“E”で指定されるフレーム間を含むGOPの番号が設定される。“JFrame”には、カーソル“E”のすぐ右側に位置するフレームの番号が設定される。“JmpFile”には、ヌル（空白文字）が設定される。

【0060】編集対象のファイル内のフレームをジャンプさせるためには、ユーザは、“ジャンプ”を選択する。これにより、カーソル43aは“J”と“D”になり、カーソル43bは無表示になる。

【0061】なお、他のファイル内のフレーム間にジャンプさせる場合、ユーザは、処理選択枝41b内の“ファイル”を選択して上記ジャンプ先のファイルを指定する。これにより、対応するファイルが編集画面に読み込まれ、そのファイルに含まれるフレームの列がフレーム表示部42bに表示される。このとき、カーソル43aは“J”になり、カーソル43bは“D”になる。

【0062】そしてユーザは、ジャンプ元となる位置にカーソル“J”を移動させ、ジャンプ先となる位置にカーソル“D”を移動させる。この後、ユーザが“ジャンプ”を再び選択すると、インデックス14（図4）内のコマンドコード18及びコマンドデータ19の設定が行われる。

【0063】すなわち、コマンドコード18は“0x0004”（図5）に設定され、コマンドデータ19につ

いては、各種の変数（図6の（d））が以下のように設定される。

【0064】変数“NumGop”には、カーソル“J”で指定されるフレーム間を含むGOPの番号が設定される。

“NumFrame”には、カーソル“J”のすぐ左側に位置するフレームの番号が設定される。“JGop”には、カーソル“D”で指定されるフレーム間を含むGOPの番号が設定される。“JFrame”には、カーソル“D”のすぐ右側に位置するフレームの番号が設定される。“JmpFile”には、ジャンプ先のファイルの名称（ファイル名）が設定される。

【0065】次に、編集対象のファイル内のフレーム間に他のファイル内のフレームを複写するためには、ユーザは、“コピー”を選択する。これにより、カーソル43aは“↑↑”となり、カーソル43bは“S”と“E”になる。次に、ユーザが処理選択枝41b内の“ファイル”を選択して上記他のファイルを指定すると、対応するファイルが編集画面に読み込まれ、そのファイルに含まれるフレームの列がフレーム表示部42bに表示される。そしてユーザは、作業エリア4bにおいて、複写元のフレーム列の始点にカーソル“S”を移動させ、終点にカーソル“E”を移動させる。また、ユーザは、作業エリア4aにおいて、上記複写先のフレームを挿入すべき位置にカーソル“↑↑”を移動させる。この後、ユーザが“コピー”を再び選択すると、インデックス14（図4）内のコマンドコード18及びコマンドデータ19の設定が行われる。

【0066】なお、このときのコマンドコード18及びコマンドデータ19は、前述の“フレーム挿入”の設定処理を行う場合のコマンドコード18及びコマンドデータ19が使用される。

【0067】したがって、コマンドコード18は“0x0002”（図5）に設定され、コマンドデータ19については、各種の変数（図6の（b））が以下のように設定される。

【0068】変数“InsGop”には、カーソル“↑↑”で指定されたフレーム間を含むGOPの番号が設定される。“InsFrame”には、カーソル“↑↑”のすぐ左側に位置するフレームの番号が設定される。“NumGop”には、カーソル“S”で指定されたフレーム間を含むGOPの番号が設定される。“NumFrame”には、カーソル“S”のすぐ左側に位置するフレームの番号が設定される。“Frames”には、カーソル“S”から“E”までの間に含まれるフレームの数が設定される。“InsFile”には、複写したファイルの名称（ファイル名）が設定される。

【0069】一方、参照ファイルを指定するためには、ユーザは、処理選択枝41a内の“参照ファイル”を選択する。なお、この“参照ファイル”は、登録後（デコード後）にユーザがそのデータを参照したいと考えるフ

10

20

30

40

50

ファイルを指定する場合に使用される。上記“参照ファイル”の選択により、編集画面上の所定の位置に、図8(a)に示すようなダイアログ50が表示される。

【0070】入力エリア51は、参照ファイル名を入力するためのエリアである。ボタン“Browse”52は、参照ファイルを検索する際に使用される。また、ボタン“OK”53は、参照ファイルの指定が終了したときに使用され、ボタン“CANCEL”54は、参照ファイルの指定をキャンセルする際に使用される。また、参照ファイルの種類(実行ファイル“EXE”，テキストファイル“TXT”の別)を指定するためには、それぞれチェック欄55、56が使用される。

【0071】この後、ユーザが“OK”を選択すると、インデックス14(図4)内のコマンドコード18及びコマンドデータ19の設定が行われる。すなわち、コマンドコード18は“0x0003”(図5)に設定され、コマンドデータ19については、各種の変数(図6の(c))が以下のように設定される。

【0072】変数“NumGop”には、コマンド“0x0003”を実行する位置を示すGOPの番号が設定される。“NumFrame”には、同様に、コマンド“0x0003”を実行する位置を示すフレームの番号が設定される。“Kind”には、参照ファイルの種類が設定される。“Insfile”には、参照ファイル名が設定される。

【0073】なお、上記説明した“ファイル挿入”，“フレーム挿入”，“カット”，“ジャンプ”，“コピー”の各々について編集作業中に、ユーザが“キャンセル”を選択すると、その作業は取り消される。

【0074】次に、上記手順で編集されたビットストリームから動画を再生する動画再生システムについて説明する。編集後のビットストリームからの動画の再生は、動画再生システムによって行われる。この動画再生システムには、デコーダ(図示せず)が設けられている。デコーダは、以下詳述するように、シーケンスヘッダ11中にインデックスコード15が設定されていれば、対応するインデックス14の内容にしたがってコマンドを実行することにより、フレームの表示順序を定めるとともに、コマンド実行後のフレームのデコード、表示等の一連の処理を行う。一方、シーケンスヘッダ11中にインデックスコード15が設定されていない場合は、デコーダは、コマンドの実行はせずに、対応するGOP12内のフレームを従来通りに順にデコード・表示する。

【0075】動画再生システムには、GOPカウンタ、フレームカウンタ、及び挿入フレームカウンタ(図示せず)が備えられる。デコーダは、再生の対象となっているビットストリームに含まれるGOPを上記GOPカウンタによって先頭から順に計数する。すなわち、GOPカウンタは、デコーダにより現在処理されているGOPの番号を識別するために使用される。なお、先頭の

GOPの番号は、GOPカウンタの計数値“0”に相当する。

【0076】また、デコーダは、GOPに含まれるフレームをフレームカウンタによって先頭から順に計数する。すなわち、フレームカウンタは、デコーダにより現在処理されているフレームの番号を識別するために使用される。なお、このときの先頭のフレームの番号は、フレームカウンタの計数値“0”に相当する。

【0077】また、挿入フレームカウンタは、後述する挿入処理の後処理において使用されるものである。すなわち、挿入フレームカウンタは、デコーダがコマンド“他のファイルの挿入”や“他のファイルの部分挿入”を実行した後に、挿入したフレームを先頭から順に計数するために使用される。

【0078】また、動画再生システムの記憶部(図示せず)には、挿入フラグが備えられる。この挿入フラグは、コマンド“ファイル挿入”や“フレーム挿入”がデコーダにより実行された場合に“1”に設定される。すなわち、挿入フラグの“1”は、挿入処理の後処理(後述)が必要な状態にあることを意味する。一方、挿入処理の後処理が必要でないときは、“0”に設定される。

【0079】なお、再生処理の開始時は、上記のGOPカウンタ、フレームカウンタ、挿入フレームカウンタ、挿入フラグの各値は“0”に設定されている。以下、図10のフローチャートを参照して、動画再生システムにおけるデコーダの動作を説明する。まず、ユーザが一番最初に再生させるべきファイル内ビットストリームをデコーダに与えると、デコーダは、そのビットストリームの先頭に設けられたシーケンスヘッダコード21(スタートコード)を検出する(ステップA1)。次に、デコーダは、上記シーケンスヘッダコード21が属するシーケンスヘッダ11の内容を図2のデータ構造にしたがって読み込む(ステップA2)。このとき、シーケンスヘッダ11の内容にしたがって各種パラメータの設定が行われる。すなわち、このとき各種パラメータに所定の値が代入される。また、フレームカウンタの値がクリアされる(“0”に設定される)。

【0080】さらに、デコーダは、ユーザデータ領域13上のインデックスコード15を検出した場合には、当該インデックスコード15の属するインデックス14内に読み込むべき内容があるものと解釈し、当該インデックス14内のすべての内容を読み込む。そして、デコーダは、検出されたインデックスコード15がシーケンスヘッダ11内に存在するものであるか否かを判断する(ステップA3)。

【0081】インデックスコード15がシーケンスヘッダ11内に存在する場合、デコーダは、GOPカウンタの値及びフレームカウンタの値がインデックス14内のインデックスGOP(IndexGop)16及びインデックスフレーム(IndexFrame)17に一致するか否かの判断を

行う(ステップA4)。

【0082】一致する場合には、デコーダは、インデックス14内に設定されているコマンドを実行する(ステップA5)。一方、一致しない場合には、何もせずにステップA6に進む。

【0083】次に、デコーダは、挿入処理の後処理を行う(ステップA6)。なお、挿入処理の後処理は、後で詳述するように、ステップA5においてコマンド“ファイル挿入”や“フレーム挿入”が実行された後に行われる処理である。

【0084】次に、デコーダは、フレームカウンタがいま表示している値に相当する1つのフレームについてデコードを行い、当該フレームをディスプレイ(図示せず)に表示する(ステップA7)。これにより、1つのフレームが再生されたことになる。この後、フレームカウンタの値を1つ加算する(ステップA8)。

【0085】次に、デコーダは、対応するGOP内のすべてのフレームのディスプレイへの表示を終えたか否かを判断する(ステップA9)。終えていなければステップA4に戻って上記の動作を繰り返し、終えていればス

テップA11に進む。

【0086】なお、上記ステップA3において、インデックスコード15がシーケンスヘッダ11内に存在しない場合には、デコーダは、対応するGOP内のすべてのフレームについてデコードを行い、これらのフレームを順にディスプレイに表示する(ステップA10)。これにより、1つのGOP内のすべてのフレームが再生されたことになる。この後、ステップA11に進む。

【0087】デコーダは、次のスタートコードの検出を行い(ステップA11)、そのスタートコードがシーケンスヘッダコード21(図2)であるか否かを判断する(ステップA12)。スタートコードがシーケンスヘッダコード21であれば、GOPカウンタの値を1つ加算してステップA2(シーケンスヘッダの読み込み処理)に戻り、上記動作を繰り返す。一方、シーケンスヘッダコード21でなければ(すなわち、シーケンスヘッダコード21が検出されなければ)、ファイルの最終位置にあると判断され、そのまま終了する。

【0088】次に、図11のフローチャートを参照して、各コマンドの実行(図10のステップA5)の詳細を説明する。デコーダは、インデックス14内に設定されているコマンドコード18の種類に応じて、各種の処理を実行する。

【0089】まず、コマンドコードが“0x0001”であるか否かが判断される(ステップB1)。コマンドコードが“0x0001”であれば、後で詳述する処理A(他のファイルの挿入)が実行されて(ステップB2)、ステップA6(図10)に戻る。

【0090】一方、ステップB1においてコマンドコードが“0x0001”でなければ、コマンドコードが

“0x0002”であるか否かが判断される(ステップB3)。コマンドコードが“0x0002”であれば、後で詳述する処理B(他のファイルの部分挿入)が実行されて(ステップB4)、ステップA6(図10)に戻る。

【0091】ステップB3においてコマンドコードが“0x0002”でなければ、コマンドコードが“0x0003”であるか否かが判断される(ステップB5)。コマンドコードが“0x0003”であれば、後

で詳述する処理C(参照ファイル指定)が実行されて(ステップB6)、ステップA6(図10)に戻る。

【0092】ステップB5においてコマンドコードが“0x0003”でなければ、コマンドコードが“0x0004”であるか否かが判断される(ステップB7)。コマンドコードが“0x0004”であれば、後で詳述する処理D(ジャンプ指定)が実行されて(ステップB8)、ステップA6(図10)に戻る。ステップB7においてコマンドコードが“0x0004”でなければ、何もせずにステップA6(図10)に戻る。

【0093】次に、図12のフローチャートを参照して、処理A(他のファイルの挿入)の詳細を説明する。まず、デコーダは、挿入フラグを“1”に設定する(ステップC1)。次に、デコーダは、挿入すべき他のファイルをこれから読み出すため、いま再生の対象となっているファイルの名称(ファイル名)を保存部(図示せず)に保存するとともに(ステップC2)、コマンドデータ(図6の(a))を保存する(ステップC3)。なお、保存したファイル名とコマンドデータは、他のファイルの挿入が完了した後、後述の挿入処理の後処理において次に表示すべきフレームを検索するために使用される。

【0094】次に、デコーダは、コマンドデータ(図6の(a))上の挿入ファイル名“InsFile”に基づき、他のファイルを読み込み(ステップC4)、そのファイルに含まれるシーケンスヘッダコード21を検索する(ステップC5)。

【0095】そして、デコーダは、検索したシーケンスヘッダコード21に基づき、シーケンスヘッダ11の内容を読み込み、各種パラメータの設定を行う(ステップC6)。この後、デコーダは、GOPカウンタとフレームカウンタの値をクリアする(“0”に設定する)(ステップC7)。そして、この処理Aを終了する。

【0096】次に、図13のフローチャートを参照して、処理B(他のファイルの部分挿入)の詳細を説明する。まず、デコーダは、挿入フレームカウンタの値をクリアする(“0”に設定する)とともに、挿入フラグを“1”に設定する(ステップD1)。次に、デコーダは、挿入すべきフレームを含む他のファイルをこれから読み出すため、いま再生の対象となっているファイルの名称(ファイル名)を保存部に保存するとともに(ステ

ップD2)、コマンドデータ(図6の(b))を保存する(ステップD3)。なお、保存したファイル名とコマンドデータは、他のファイルのフレームの挿入が完了した後、後述の挿入処理の後処理において次に表示すべきフレームを検索するために使用される。

【0097】次に、デコーダは、コマンドデータ(図6の(b))上の挿入ファイル名“InsFile”に基づき、他のファイルを読み込む(ステップD4)。デコーダは、上記他のファイル上における表示の開始位置を検索するため、まず、コマンドデータ(図6の(b))上の挿入GOP番号“NumGop”と挿入フレーム番号“NumFrame”をそれぞれ記憶部に記憶する(ステップD5)。

【0098】そして、デコーダは、後で詳述する表示開始位置の検索処理を行う(ステップD6)。なお、上記の記憶した挿入GOP番号と挿入フレーム番号は、表示開始位置の検索処理において使用され、それぞれ検索GOP番号、検索フレーム番号と呼ばれる。そして、この処理Bを終了する。次に、図14のフローチャートを参照して、処理C(参照ファイル指定)の詳細を説明する。デコーダは、ここでコマンドの登録処理を行う(ステップE1)。このコマンドの登録処理においては、コマンドコード“0x0003”が出現する順番に登録が行われる。図8(b)に示すように、各種のデータ(GOP番号、フレーム番号、ファイルの種類、ファイル名)が上から順に登録されていく様子が示される。これにより、ユーザは、登録された各種のデータについて参照することが可能となる。上記コマンドの登録処理が完了すると、処理Cが終了する。

【0099】なお、ファイルの先頭において、全てのコマンドコード“0x0003”の内容を初めから登録しないのは、ユーザが映像を見ていて途中でメモ等を書き込みたいときに、映像を止めてメモを書き込んだデータを別のファイルとして作成し、作成したファイル名及びその位置を記録するためである。

【0100】次に、図15のフローチャートを参照して、処理D(ジャンプ指定)の詳細を説明する。まず、デコーダは、コマンドデータ(図6の(d))上のジャンプ先のファイル名“JmpFile”に基づき、他のファイルを読み込む(ステップF1)。なお、“JmpFile”がヌル(=0)に設定されている場合、読み込みは行われない。

【0101】次に、デコーダは、上記他のファイル上における表示の開始位置を検索するため、まず、コマンドデータ(図6の(d))上のジャンプ先のGOP番号“JGop”とジャンプ先のフレーム番号“JFrame”をそれぞれ記憶部に記憶する(ステップF2)。

【0102】そして、デコーダは、後で詳述する表示開始位置の検索処理を行う(ステップF3)。なお、上記の記憶したジャンプ先のGOP番号とジャンプ先のフレーム番号は、表示開始位置の検索処理において使用さ

れ、それぞれ検索GOP番号、検索フレーム番号と呼ばれる。

【0103】この後、デコーダは、GOPカウンタの値をコマンドデータ(図6の(d))上のジャンプ先のGOP番号“JGop”と同値に設定するとともに、フレームカウンタの値をジャンプ先のフレーム番号“JFrame”と同値に設定して(ステップF4)、この処理Dを終了する。

【0104】次に、図16のフローチャートを参照して、挿入処理の後処理(図10のステップA6)の詳細を説明する。まず、デコーダは、挿入フラグが“1”を示しているか否かを判断する(ステップG1)。挿入フラグが“1”を示していなければ、この挿入処理の後処理を終える。一方、挿入フラグが“1”を示していれば、挿入フレームカウンタの値が1つ加算される(ステップG2)。

【0105】いま行っている後処理が“フレーム挿入”の後処理である場合には、デコーダは、挿入フレームカウンタの値とコマンドデータ19内の挿入フレーム数“Frames”(図6の(b))との比較を行う(ステップG3)。比較した結果が一致を示す場合には、デコーダは、すべての挿入フレームの表示が終了したものと見なし、すでに保存部に保存してあるファイル名に基づき、対応するファイルを読み込む(ステップG4)。一方、比較した結果が一致を示さない場合には、何もせずにこの挿入処理の後処理を終える。

【0106】なお、上記ステップG3において、いま行っている後処理が“ファイル挿入”の後処理である場合には、デコーダは、挿入フレームカウンタの値が挿入ファイルの最終位置を示しているか否かを判別する(図示せず)。最終位置を示していれば、デコーダは、挿入ファイルの表示が終了したものと見なし、ステップG4に進む。一方、最終位置を示していなければ、何もせずにこの挿入処理の後処理を終える。

【0107】次に、デコーダは、読み込んだファイル上における表示の開始位置を検索するため、まず、すでに保存部に保存してあるコマンドデータ内の挿入先のGOP番号“InsGop”と挿入先のフレーム番号“InsFrame”を記憶部に記憶する(ステップG5)。

【0108】そして、デコーダは、後で詳述する表示開始位置の検索処理を行う(ステップG6)。なお、上記の記憶した挿入先のGOP番号と挿入先のフレーム番号は、表示開始位置の検索処理で使用される際に、それぞれ検索GOP番号、検索フレーム番号と呼ばれる。そして、挿入フラグの値をクリアした後(“0”に設定した後)、この挿入処理の後処理を終える。

【0109】次に、図17のフローチャートを参照して、表示開始位置の検索処理(図13のステップD6、図15のステップF3、及び図16のステップG6)の詳細を説明する。

【0110】まず、デコーダは、GOPカウンタの値が検索GOP番号に一致するか否かを判断する(ステップH1)。ここで一致しなければ、デコーダは、シーケンスヘッダコード21の検出を行い(ステップH2)、対応するシーケンスヘッダ11を読み込む(ステップH3)。次に、デコーダは、GOPカウンタの値を1つ加算するとともに、フレームカウンタをクリアする(“0”に設定する)(ステップH4)。そして、ステップH1に戻る。

【0111】ステップH1において、GOPカウンタの値が検索GOP番号に一致すれば、デコーダは、そのGOPの中に表示を開始すべき位置が存在するものと見なす。次に、デコーダは、フレームカウンタの値と検索フレーム番号とが一致するか否かを判断する(ステップH5)。ここで一致しなければ、デコーダは、フレームカウンタが示すフレームの再生を行う。ただし、このときに再生するフレームは表示しない(ステップH6)。

【0112】そして、デコーダは、フレームカウンタの値を1つ加算し(ステップH7)、ステップH5に戻る。ステップH5において、フレームカウンタの値が検索フレーム番号に一致すれば、そのフレームの存在する位置が表示を開始すべき位置であるものとみなし、この表示開始位置の検索処理を終了する。

【0113】以上説明したように上記実施例によれば、ユーザが編集画面において“ファイル挿入”、“フレーム挿入”、“カット”、“ジャンプ”、“コピー”等の作業をすることにより、ユーザデータ領域13内のインデックス中に、フレームの表示順序を定義する情報が自動的に作成される。すなわち、フレーム単位での細かいビットストリームの編集を容易に行うことができる。また、こうして細かく編集されたビットストリームから繊細な動画を容易に再生することができる。

【0114】なお、本実施例では、MPEG1方式のビットストリームを採用した場合を説明したが、これに限定されることなく、他の方式のビットストリームにも適用可能である。

【0115】

【発明の効果】以上説明したように、本発明に係る動画編集システムによれば、静止画像単位での細かい動画データデータの編集を容易に行うことができる。また、本発明に係る動画再生システムによれば、静止画像単位で編集された動画データから繊細な動画を容易に再生することができる。

【図面の簡単な説明】

【図1】本発明の一実施例に係る動画編集システムで編集するビットストリームの構成を示す図。

【図2】同実施例のビットストリームに含まれるシーケンスヘッダのデータ構造を示す図。

【図3】同実施例のビットストリームに含まれるGOPのデータ構造を示す図。

【図4】同実施例のシーケンスヘッダのユーザデータ領域に格納されるインデックスのデータ構造を示す図。

【図5】同実施例のインデックスに格納されるコマンドコードの種類を示す図。

【図6】同実施例のインデックスに格納されるコマンドデータのデータ構造を示す図。

【図7】同実施例における動画編集システムの編集画面を示す図。

【図8】同実施例における参照ファイルに関するダイアログ及び登録手順を示す図。

【図9】同実施例の編集画面で使用されるフレーム指示カーソルの種類を示す図。

【図10】同実施例の動画再生システムにおけるデコーダの動作を示すフローチャート。

【図11】同実施例のデコーダによる各コマンドの処理を示すフローチャート。

【図12】同実施例のデコーダによる処理Aを示すフローチャート。

【図13】同実施例のデコーダによる処理Bを示すフローチャート。

【図14】同実施例のデコーダによる処理Cを示すフローチャート。

【図15】同実施例のデコーダによる処理Dを示すフローチャート。

【図16】同実施例のデコーダによる挿入処理の後処理を示すフローチャート。

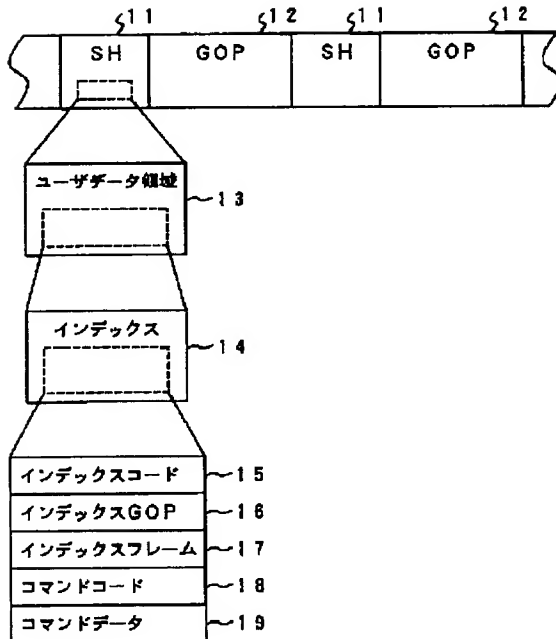
【図17】同実施例のデコーダによる表示開始位置の検索処理を示すフローチャート。

【図18】従来の動画編集システムで使用されるビットストリームの構成を示す図。

【符号の説明】

1, 11…シーケンスヘッダ、2, 12…GOP、13…ユーザデータ領域、14…インデックス、15…インデックスコード、16…インデックスGOP、17…インデックスフレーム、18…コマンドコード、19…コマンドデータ、21…シーケンスヘッダコード、22…水平サイズ、23…垂直サイズ、24…画素アスペクト比、25, 35…ユーザデータ、31…グループスタートコード、32…タイムコード、33, 34…フラグ、4a, 4b…作業エリア、41a, 41b…処理選択枝、42a, 42b…フレーム表示部、43a, 43b…カーソル、44a, 44b, 45a, 45b…スクロールボタン、46a, 46b…表示エリア、47a, 47b…スライダ、50…ダイアログ、51…入力エリア、52~54…ボタン、55, 56…チェック欄。

【図1】



【図5】

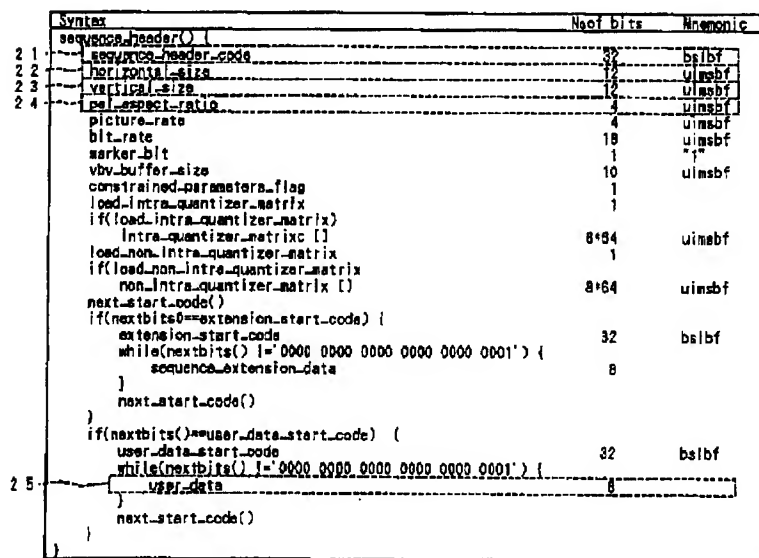
コマンドコード	意味
0×0001	他のファイルの挿入
0×0002	他のファイルの部分挿入
0×0003	参照ファイル指定
0×0004	ジャンプ指定

【図9】

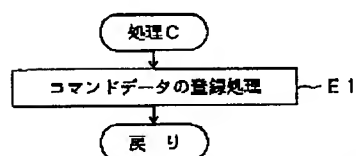
編集コマンド	カーソル	カーソルの示す意味
ジャンプ	J	ジャンプ開始位置
	D	ジャンプ先の位置
コピー、カット	S	コピー、カット開始位置
	E	コピー、カット終了位置
フレーム挿入、 ファイル挿入	↑↑, SE	位置

【図11】

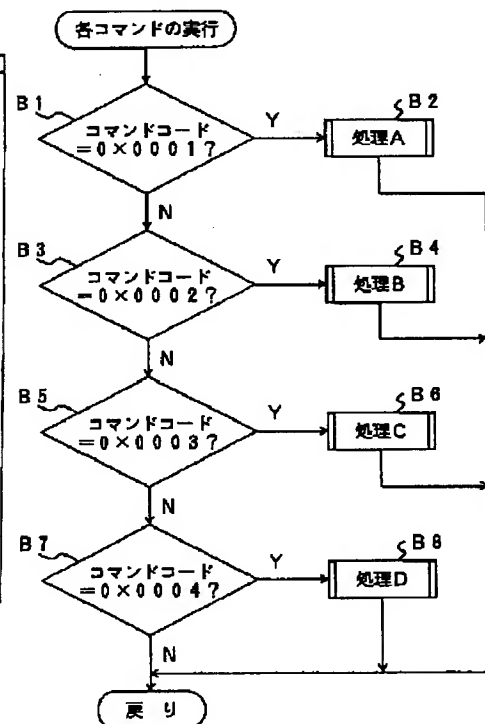
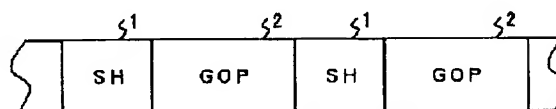
【図2】



【図14】



【図18】



【図3】

Syntax	Next bits	Mnemonic
group_of_picture() {		
31 group_start_code	92	bslbf
32 time_code	25	
33 closed_gop	1	
34 broken_link	1	
next_start_code()		
if(nextbits()==extension_start_code) {		
extension_start_code	32	bslbf
while(nextbits() != '0000 0000 0000 0000 0001') {		
group_extension_data	8	
}		
next_start_code()		
}		
if(nextbits()==user_data_start_code) {		
user_data_start_code	32	bslbf
while(nextbits() != '0000 0000 0000 0000 0001') {		
user_data	8	
}		
next_start_code()		
}		
do {		
picture()		
} while(nextbits()==picture_start_code)		
}		

【図4】

14

5

```
while(nextbits()=='0000 0000 0000 0001') {
```

IndexCode

18bit

IndexGop

32bit

IndexFrame

32bit

CommandCode

18bit

CommandData

n byte

}

【図8】

§ 50

参照ファイル名 § 51 Browse § 52

ファイルの種類

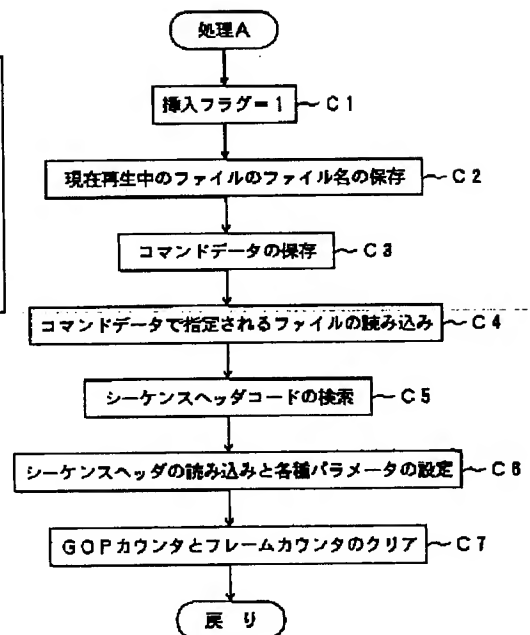
55 ☐ 実行ファイル OK § 53

56 ☒ テキストファイル CANCEL § 54

(b)

GOP番号	フレーム番号	ファイルの種類	ファイル名
10	14	TXT	text1.txt
40	20	EXE	exe1.exe
⋮	⋮	⋮	⋮

【図12】



【図6】

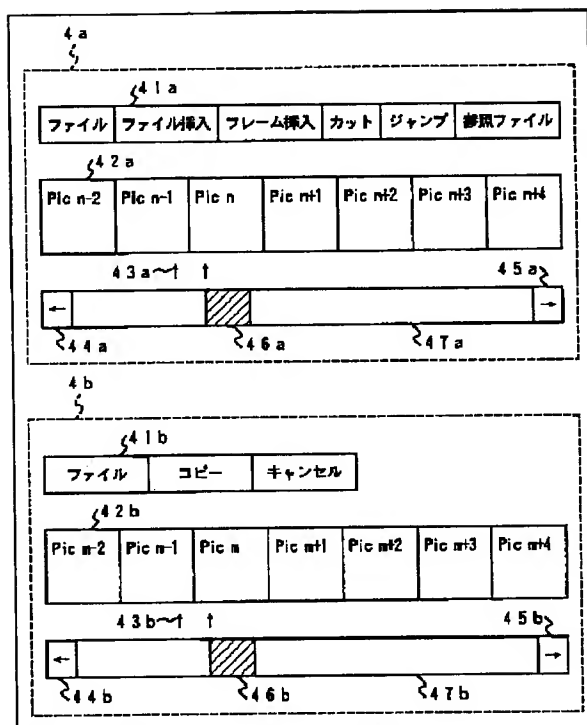
(a)		
コマンド	他のファイルの挿入	
コマンドコード	0×0001	
コマンドデータの構造		
変数名	大きさ	意味
InsGop	32 bit	挿入先のGOP番号
InsFrame	32 bit	挿入先のフレーム番号
InsFile	128 byte	挿入ファイル名

(b)		
コマンド	他のファイルの部分挿入	
コマンドコード	0×0002	
コマンドデータの構造		
変数名	大きさ	意味
InsGop	32 bit	挿入先のGOP番号
InsFrame	32 bit	挿入先のフレーム番号
NumGop	32 bit	挿入GOP番号
NumFrame	32 bit	挿入フレーム番号
Frames	32 bit	挿入フレーム数
InsFile	128 byte	挿入ファイル名

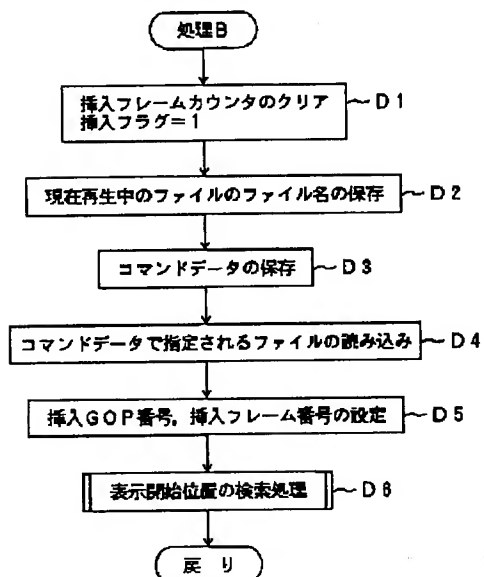
(c)		
コマンド	参照ファイル指定	
コマンドコード	0×0003	
コマンドデータの構造		
変数名	大きさ	意味
NumGop	32 bit	GOP番号
NumFrame	32 bit	フレーム番号
Kind	4 byte	参照ファイルの種類
InsFile	128 byte	参照ファイル名

(d)		
コマンド	ジャンプ指定	
コマンドコード	0×0004	
コマンドデータの構造		
変数名	大きさ	意味
NumGop	32 bit	GOP番号
NumFrame	32 bit	フレーム番号
JGop	32 bit	ジャンプ先のGOP番号
JFrame	32 bit	ジャンプ先のフレーム番号
JumpFile	128 byte	ジャンプ先のファイル名

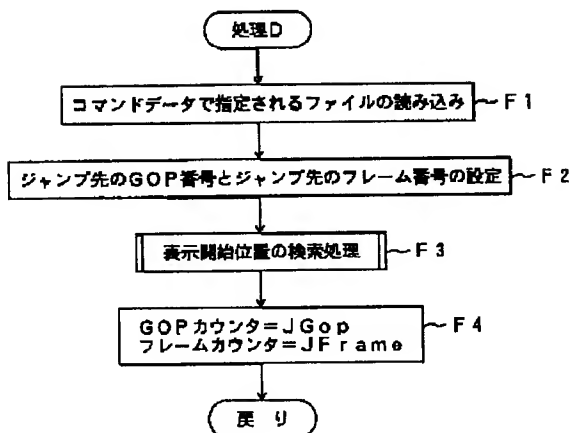
【図7】



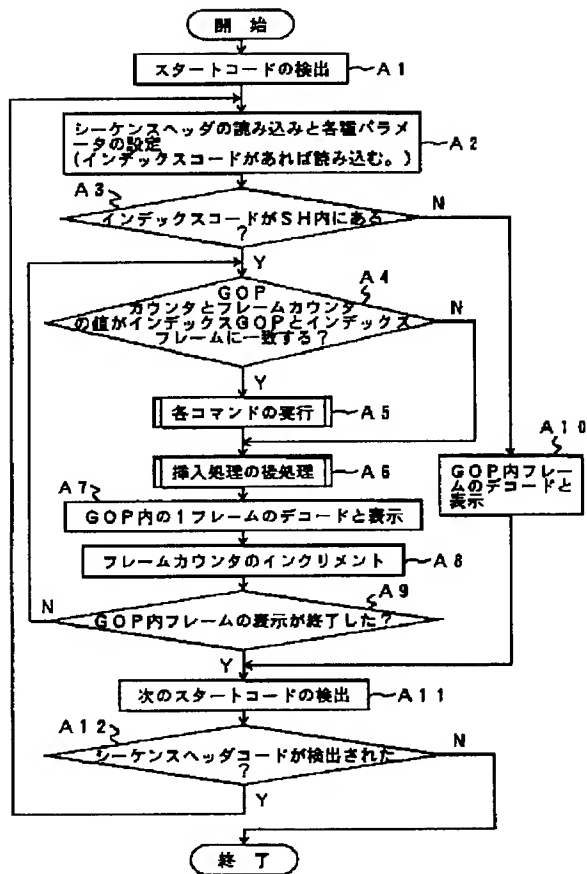
【図13】



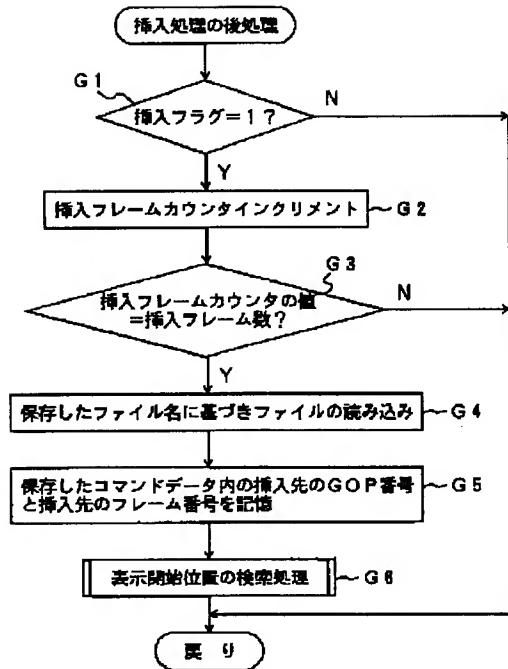
【図15】



【図10】



【図16】



【図17】

